IN THE CLAIMS

Listing of Claims:

Claim 1 (currently amended) A method for stack memory protection comprising the steps of:

generating new memory page attributes for a page table used to manage memory, each of said new memory page attributes identifying a block of memory as a new class of memory, each of said new memory page attributes generated by a corresponding new load/store instruction;

assigning, by an operating system or a processor, a selected one of said new memory page attributes to a selected block of memory, said selected block of memory used as a new class of memory corresponding to said selected new memory page attribute;

blocking normal load /stores to a memory block having one of said new memory page attributes; and

blocking a first new load/store to a memory block with one of said new memory page attributes not corresponding to said first new load/store.

Claim 2 (currently amended) The method of claim 1, wherein said new classes of memory comprise stack memory with corresponding stack memory load/store instructions.

Claim 3 (currently amended) The method in claim 2, wherein a first error condition is generated whenever normal load/stores are attempted to stack memory having a first or a second stack memory attribute with corresponding first or second stack memory load/store instructions.

Claim 4 (currently amended) The method in claim 2, wherein a second error condition is generated whenever [[said]] stack memory load/stores are attempted to said memory not having said first or second stack memory attribute.

Claim 5 (original) The method in claim 2, wherein a third error condition is generated whenever a stack memory load/store for a first memory stack is attempted to a second memory stack, said third error condition also generated if a stack memory load/store for said second memory stack is attempted to said first memory stack.

Claim 6 (original) The method of claim 2, wherein said stack memory load/store instructions are executed on a CPU comprising an IA64 architecture.

Claim 7 (currently amended) The method of claim 5, wherein said first memory stack is a processor stack, said processor stack used by a processor to load and store hardware register contents during program execution, said processor stack [[stacks]] is transparent to a programmer or a compiler.

Claim 8 (original) The method of claim 7, wherein said processor stack is an IA64 register stack.

Claim 9 (original) The method of claim 5, wherein said second memory stack is a program stack, said program stack used by a programmer or a compiler in managing program flow.

Claim 10 (original) A processor comprising stack memory protection circuitry, said processor using blocks of memory as stack memory, said stack memory protection circuitry comprising:

a stack memory attribute circuit, said stack memory attribute circuit operable to generate memory attributes, said memory attributes associated with each memory block designated as a memory stack;

a page table attribute storage circuit, said page table attribute circuit operable to store and associate one of said stack memory attributes with a block of memory designated as stack memory;

a stack memory allocation circuit, said stack memory allocation circuit operable to identify a block of memory as a stack memory and associate said memory block with one of said stack memory attributes, said stack memory attributes stored in a memory page table; and

a stack memory instruction execution circuit, said stack memory instruction execution circuit operable to decode load/store instructions to memory blocks, said stack memory instruction execution circuit granting stack memory load and stores to memory blocks having a required stack memory attribute and not granting stack memory load and stores to memory blocks not having said required stack memory attribute.

Claim 11 (original) The processor in claim 10, wherein a first error condition is generated whenever normal load/stores are attempted to stack memory having a first or a second stack memory attribute.

Claim 12 (original) The processor in claim 10, wherein a second error condition is generated whenever said stack memory load/stores are attempted to memory not having a stack memory attribute.

Claim 13 (original) The processor in claim 10, wherein a third error condition is generated whenever a stack memory load/store for a first memory stack is attempted to a second memory stack, said third error condition also generated if a stack memory load/store for said second memory stack is attempted to said first memory stack.

Claim 14 (original) The processor of claim 10, wherein said stack memory load and store instructions are executed on a CPU comprises an IA64 architecture.

Claim 15 (original) The processor of claim 13, wherein said first memory stack is a processor stack, said processor stack used by a processor to load and store hardware register contents during program execution, said processor stacks transparent to a programmer or a compiler.

Claim 16 (original) The processor of claim 13, wherein said second memory stack is a program stack, said program stack used by a programmer or a compiler in managing program flow.

Claim 17 (original) A data processing system, comprising:

a central processing unit (CPU);

shared random access memory (RAM);

read only memory (ROM);

an I/O adapter; and

a bus system coupling said CPU to said ROM, said RAM said display adapter, wherein said CPU, said CPU comprising stack memory protection circuitry, said stack memory protection circuitry comprising:

a stack memory attribute circuit, said stack memory attribute circuit operable to generate memory attribute, said memory attribute associated with each memory block designated as a memory stack;

a page table attribute storage circuit, said page table attribute circuit operable to store and associate said stack memory attribute with a block of memory designated as stack memory;

a stack memory allocation circuit, said stack memory allocation circuit operable to identify a block of memory as a stack memory and associate said memory block with a stack memory attribute, said stack memory attribute stored in a memory page table; and

a stack memory instruction execution circuit, said stack memory instruction execution circuit operable to decode load/store instructions to memory blocks, said stack memory instruction execution circuit granting stack memory load and stores to memory blocks having a stack memory attribute and not granting stack memory load and stores to memory blocks not having said stack memory attribute.

Claim 18 (original) The data processing system in claim 17, wherein a first error condition is generated whenever normal load/stores are attempted to stack memory having a first or a second stack memory attribute.

Claim 19 (original) The data processing system in claim 17, wherein a second error condition is generated whenever said stack memory load/stores are attempted to memory not having a stack memory attribute.

Claim 20 (original) The data processing system in claim 17, wherein a third error condition is generated whenever a stack memory load/store for a first memory stack is attempted to a second memory stack, said third error condition also generated if a stack memory load/store for said second memory stack is attempted to said first memory stack.

Claim 21 (original) The data processing system of claim 17, wherein said stack memory load and store instructions are executed on a CPU comprising an IA64 architecture.

Claim 22 (original) The data processing system of claim 20, wherein said first memory stack is a processor stack, said processor stack used by a processor to load and store hardware register contents during program execution, said processor stacks transparent to a programmer or a compiler.

Claim 23 (original) The data processing system of claim 20, wherein said second memory stack is a program stack, said program stack used by a programmer or a compiler in managing program flow.

Claim 24 (currently amended) A computer program product embodied in a machine readable medium, including an operating system and a compiler for a processor system, comprising; a program of instructions for performing the program steps of:

generating new memory page attributes for a page table used to manage memory, each of said new memory page attributes identifying a block of memory as a new class of memory, each of said new memory page attributes generated by a corresponding new load/store instruction;

assigning, by an operating system or a processor, a selected one of said new memory page attributes to a selected block of memory, said selected block of memory used as a new class of memory corresponding to said selected new memory page attribute;

blocking normal load /stores to a memory block having one of said new memory page attributes; and

blocking a first new load/store to a memory block with one of said new memory page attributes not corresponding to said first new load/store.

Claim 25 (original) The computer program product of claim 24, wherein said new classes of memory comprise stack memory.

Claim 26 (original) The computer program product in claim 25, wherein a first error condition is generated whenever normal load/stores are attempted to stack memory having a first or a second stack memory attribute.

Claim 27 (original) The computer program product in claim 25, wherein a second error condition is generated whenever said stack memory load/stores are attempted to memory not having said stack memory attribute.

Claim 28 (original) The computer program product in claim 25, wherein a third error condition is generated whenever a stack memory load/store for a first memory stack is attempted to a second memory stack, said third error condition also generated if a stack memory load/store for said second memory stack is attempted to said first memory stack.

Claim 29 (original) The computer program product of claim 25, wherein said stack memory load/store instructions are executed on a CPU comprising an IA64 architecture.

Claim 30 (original) The computer program product of claim 29, wherein said first memory stack is a processor stack, said processor stack used by a processor to load and store hardware register contents during program execution, said processor stacks transparent to a programmer or a compiler.

Claim 31 (original) The computer program product of claim 30, wherein said processor stack is an IA64 register stack.

Claim 32 (original) The computer program product of claim 28, wherein said second memory stack is a program stack, said program stack used by a programmer or a compiler in managing program flow.

Claim 33 (original) A method of managing a memory device comprising the steps of:

partitioning said memory device into a plurality of memory spaces on an as-needed basis; and

associating a memory attribute with each memory space; said memory attribute determining a use of each of said memory spaces.

Claim 34 (original) The method of claim 33, wherein a particular memory attribute has corresponding load/store instruction.

Claim 35 (original) The method of claim 34, wherein a load/store instruction associated with a first memory attribute causes an error condition if attempted on a memory space with a second memory attribute.

Claim 36 (original) The method of claim 33, wherein each of said memory attributes are stored in a memory page table, said memory page table used to manage said memory device.